REMARKS

Claims 1-34 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 102(b) Rejection:

The Office Action rejected claims 1-9, 11-19, 21-29 and 31-34 under 35 U.S.C. § 102(b) as being anticipated by Krishnan ("Heap: Pleasures and Pains", Microsoft Corporation, February 1999, pp. 1-6). Applicants respectfully submit that claims 1-9, 11-19, 21-29 and 31-34 are not anticipated by Krishnan.

Regarding claim 1, Krishnan fails to teach executing a process within the virtual machine, wherein the virtual machine provides a platform-independent operating environment on a particular computer platform, wherein the virtual machine comprises a virtual machine virtual memory manager. The Examiner has cited a single sentence in Krishnan (top of page 2) where Krishnan states: "Languages like Microsoft Visual Basic and Java also offer new operators and use garbage collection instead of heaps." (emphasis added). This sentence, in the midst of a discussion regarding how the C/C++ Run Time Allocator works, simply implies that Java uses garbage collection instead of heaps and thus implies that Krishnan's recommendations on how to avoid performance issues related to heap management under MS Windows do not apply to languages like Java. The Examiner contends that this single mention of Java infers that Krishnan's other teachings apply equally to MS Windows heap management and virtual memory management on virtual machines. This is clearly incorrect. Krishnan mentions Java specifically to point out that the common heap problems and solutions discussed by Krishnan do not apply to Java because Java uses garbage collection instead of heaps. Therefore, Krishnan clearly fails to disclose anything regarding a virtual machine that comprises a virtual machine virtual memory manager.

Additionally the Examiner contends that MS Windows Virtual Memory Allocator, as discussed by Krishnan, is a virtual machine virtual memory manager. The Virtual Memory Allocator shown in the diagram on p. 1 of Krishnan is part of the Windows NT operating system, which is not a virtual machine that provides a platform-independent operating environment on a particular computer platform. It is well known that the Virtual Memory Allocator of MS Windows is not a *virtual machine* virtual memory manager. The Examiner is merely attempting in hindsight to forcefully insert limitations from Applicants' claim 1 into Krishnan's teachings through a single reference to the Java language without any other mention in Krishnan regarding virtual machines or virtual machine virtual memory managers. In fact, Krishnan's entire discussion relates to a heap that "sits on top of the operating system's Virtual Memory Manager" (Krishnan, page 2, paragraph 8). Even if a Java Virtual Machine (JVM) was running on a Windows platform, the Virtual Memory Allocator described in Krishnan would still be part of the Windows operating system, not the JVM. Krishnan's Virtual Memory Allocator has nothing to do with a virtual machine that provides a platform-independent operating environment on a particular computer platform.

Krishnan further fails to disclose the virtual machine virtual memory manager copying a section of the store heap including the first object to an in-memory heap in response to the process referencing the first object. The Examiner does not cite any portion of Krishnan that teaches this and Krishnan fails to mention anything about a virtual machine virtual memory manager copying a section of the store heap in response to the process referencing the first object. In contrast, Krishnan discusses common problems with heap management, such as performance slowdowns resulting from allocation operations, free operations, heap contention, heap corruption and reallocations. Krishnan also discusses various techniques programmers can use to prevent such common problems. Thus, Krishnan is concerned with heap issues and does not discuss virtual memory management at all. Plus, as noted above, MS Windows heap management and virtual memory management have nothing whatsoever to do with virtual machines. Krishnan thus clearly fails to teach the virtual machine virtual memory

manager copying a section of the store heap including the first object to an in-memory heap in response to the process referencing the first object.

The Examiner also refers to a detailed description on Virtual Memory Manager in Window NT in a separate document (Kath, "The Virtual-Memory Manager in Windows NT"). However, as with Krishnan, the Kath reference discusses virtual memory management of MS Windows, and as noted above, virtual memory management under MS Windows has nothing to do with, nor does the Kath reference mention, virtual machines or virtual machine virtual memory management.

Thus, claim 1 is clearly not anticipated by Krishnan and removal of the 102(b) rejection is respectfully requested. Similar remarks as those above regarding claim 1 also apply to claims 15 and 25.

Regarding claim 2, Krishnan fails to teach the virtual machine virtual memory manager replacing the section of the store heap with the copy of the section from the in-memory heap including the first object in response to said modifying the first object in the in-memory heap. The Examiner has failed to cite any particular portion of Krishnan regarding claim 2, but merely states that claim 2 is "rejected using the same rationale as set forth above with respect to claim 1." However, Krishnan does not discuss anything regarding a virtual machine virtual memory manager replacing a section of a store heap in response to a process modifying an object in an in-memory heap. As discussed above regarding claim 1, Krishnan discusses only certain performance problems associated with MS Windows heap management, but fails to discuss a virtual memory manager except to point out that the heap in MS Windows sits above the MS Windows virtual memory manager (Krishnan, page 2, paragraph 8). Additionally, it is also well-known that the virtual memory manager under MS Windows does not replace a section of a store heap with a copy of the section from a in-memory heap in response to a process modifying an object in the in-memory heap. In contrast, (as pointed out in the Kath reference introduced by the Examiner) the MS Windows Virtual Memory Manager only updates a memory page associated with one process in response to running out of memory for a

different process and not in response to a process modifying an object in an in-memory heap. Thus, the rejection of claim 2 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 2 also apply to claims 16 and 26.

Regarding claim 3, Krishnan fails to teach the virtual machine virtual memory manager <u>removing the copy of the section</u> from the in-memory heap after said replacing the section of the store heap. As noted above, Krishnan fails to mention anything regarding a virtual machine virtual memory manager. In addition, Krishnan does not disclose removing the copy of the section from the in-memory heap after replacing the section of the store heap. Instead Krishnan teaches various techniques for avoiding common pitfalls related to heap usage under MS Windows.

Regarding claim 11, Krishnan fails to teach wherein the memory device comprising the store heap is <u>coupled to the device via the Internet</u> so that the virtual machine virtual memory manager <u>copying the section of the store heap to the in-memory heap occurs over the Internet</u>. The Examiner fails to cite any passage of Krishnan, but instead just states that claim 11 is "rejected using the same rationale as set forth above with respect to claim 1." However, Krishnan fails to mention anything regarding a memory device comprising the store heap being coupled to the device via the Internet. In fact, Krishnan doesn't mention the Internet at all. Nor does Krishnan teach anything about multiple devices. Krishnan limits his discussion to how MS Windows heap management works and MS Windows heap management cannot copy sections of a store heap to an in-memory heap over the Internet. Krishnan clearly fails to anticipate claim 11 and removal of this rejection is respectfully requested. Similar arguments as those above regarding claim 11 also apply to claims 21 and 31.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion in regard to the dependent claims is not necessary at this time.

<u>Section 103(a) Rejection</u>:

The Office Action rejected claims 10, 20 and 30 under 35 U.S.C. § 103(a) as being unpatentable over Krishnan in view of Sukegawa (U.S. Patent 5,860,083). This rejection is unsupported by the cited art for at least the reasons given above in regard to the respective independent claims. Furthermore, the flash memory cache described in Sukegawa is cache for a hard disk drive (HDD) and has nothing to do with cache lines for a virtual memory store heap and sections thereof in an in-memory heap as recited in claims 10, 20 and 30. Although flash memory may be known to be advantageous <u>in other contexts</u>, the Examiner has not cited any reference that suggests the use of flash memory for a store heap of a virtual machine.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above referenced application from becoming abandoned, Applicants hereby petition for such extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-49700/RCK.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

☐ Petition for Extension of Time

☐ Notice of Change of Address

☐ Fee Authorization Form authorizing a deposit account debit in the amount of $ for fees ( ).

☐ Other:

Respectfully submitted,

Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: October 15, 2004